





Substituindo a biblioteca Material-UI pela Bootstrap nas primeiras telas codificadas do Projeto Integrado.

O React Js é uma biblioteca focada no desenvolvimento de interfaces para sistemas Web. Dentre as suas características, uma das mais importantes é a fácil integração com outras bibliotecas. Nesse sentido, podemos criar toda a interface de nossa aplicação apenas com os componentes nativos do React, ou podemos aprimorar essas telas com o uso de bibliotecas de terceiros. O uso de uma ou mais bibliotecas, formando um código mesclado com os componentes nativos do React, é algo muito comum. Entretanto, essa prática requer atenção pois, muitas vezes, um componente de uma biblioteca possui o mesmo nome de outro componente, em outra biblioteca. Quando isso acontece, temos um conflito, levando a aplicação a apresentar um comportamento não desejado. Um bom exemplo disso é o componente Modal, da biblioteca Material-UI. Na biblioteca Bootstrap, que veremos mais a fundo nesse módulo, também há um componente com o mesmo nome. Portanto, caso você utilize essas duas bibliotecas em seu projeto, cuide para que não aconteçam conflitos no uso do componente em questão, ou de outros que possam ser comuns, nessas ou em outras bibliotecas.

Após conhecermos e utilizarmos a biblioteca Material-UI, veremos agora outra biblioteca, que está entre as mais utilizadas n desenvolvimento Web, uma vez que possui versões JS e CSS, usadas em páginas web comuns, como versões específicas para outras bibliotecas, como o React. Trata-se da biblioteca Bootstrap -



sendo que, ao longo desse módulo, veremos mais a React Bootstrap (que, segundo o site oficial, trata-se do "framework front-end mais popular), versão voltada para o React. Caso você já tenha experiência com a Bootstrap, perceberá que o uso da segunda é muito parecido ao da primeira.

Para começar, precisaremos instalar o framework. Nesse caso, recomendo que você crie um novo projeto React, instalando a biblioteca a seguir. Esse processo pode ser visto abaixo:

```
npx create-react-app template-bootstrap
cd template-bootstrap
npm install react-bootstrap bootstrap
```

OBS: caso prefira, poderá utilizar o yarn para instalar a react-bootstrap.

Após a instalação, precisaremos realizar mais um passo para deixar a biblioteca pronta para uso. No arquivo src/index.js ou no src/App.js, inclua a linha abaixo:

```
{/* A linha abaixo deverá ser inserida em src/index.js ou src/App.js*/}
import 'bootstrap/dist/css/bootstrap.min.css';
```

Pronto, agora a biblioteca encontra-se pronta para ser utilizada. Logo, basta importar o componente que você deseja e usá-lo em seus componentes. E, por falar em componentes, a lista aqui também é extensa. Veja alguns exemplos de componentes prontos para o uso – ou, "plug and play":

- Alerts (os famosos alertas Javascript, numa versão aprimorada);
- Accordion (utilizado para expandir e retrair conteúdos em uma página);
- Buttons;
- Cards (muito úteis para a exibição de itens compostos por título, imagem e descrição, em uma lista, por exemplo);
- Navbar;
- Table;
- Forms;
- Entre outros.

Existem componentes para diversas finalidades. Além disso, o Bootstrap é conhecido pelo seu sistema de grid, a partir do qual podemos criar páginas responsivas, definindo contêineres, configurando suas larguras e, com isso, permitindo que a página responda a diferentes tamanhos de tela, reorganizando os elementos nela contidos.

A tabela abaixo demonstra os “breakpoints” do grid Bootstrap, que são seis ao total, embora você possa criar outros, customizad

BREAKPOINT	CLASSE	DIMENSÕES
X-SMALL	--	< 576px
SMALL	sm	>= 576px
MEDIUM	md	>= 768px
LARGE	lg	>= 992px
EXTRA LARGE	xl	>= 1200px
EXTRA EXTRA LARGE	xxl	>= 1400px

Tabela 1: Breakpoints do Bootstrap

Em complemento aos breakpoints, temos a Grid do Bootstrap. Aqui, uma série de componentes são disponibilizados para organizar o layout, num esquema composto por linhas e colunas. Veja o código abaixo para conhecer essa dinâmica:

```
import Container from 'react-bootstrap/Container';
import Row from 'react-bootstrap/Row';
import Col from 'react-bootstrap/Col';

const ExemploGrid() {
  return (
    <Container fluid>
      <Row>
        <Col>Coluna 1 de 1</Col>
      </Row>
    </Container>
  );
}

export default ExemploGrid;
```

Além do que vimos acima, podemos ter inúmeras combinações, variando a quantidade de linhas e colunas. Também podemos combinar as classes de breakpoint vistas anteriormente, modificando assim o comportamento do layout da Grid, composto pelas linhas e pelas colunas.

Conhecer os breakpoints e a grid do bootstrap é importante para sabermos como organizar o layout de nossa aplicação, assim como qual ou quais classes devemos utilizar. Você pode aprofundar seu entendimento sobre esse tema, lendo mais a respeito e/ou fazendo testes no seu código e vendo o comportamento de cada classe na prática.

Além dos breakpoints, e como já mencionado acima, é importante conhecer alguns componentes do framework. Para o nosso caso, uma vez que desejamos criar o frontend de diversos CRUDs, um componente primordial é o Form.

Veja o código abaixo, onde um formulário simples, composto por Email e Senha, é codificado com o react-bootstrap e o componente Form:



```
import Button from 'react-bootstrap/Button';
import Form from 'react-bootstrap/Form';

const FormularioSimples = () => {
  return (
    <Form>
      <Form.Group className="mb-3" controlId="formBasicEmail">
        <Form.Label>E-mail </Form.Label>
        <Form.Control type="email" placeholder="E-mail" />
        <Form.Text className="text-muted">
          Não compartilhe seus dados com ninguém.
        </Form.Text>
      </Form.Group>

      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Senha</Form.Label>
        <Form.Control type="password" placeholder="Senha" />
      </Form.Group>
      <Button variant="primary" type="submit">
        Submit
      </Button>
    </Form>
  );
}

export default FormularioSimples;
```

Analisando o código, podemos perceber que o componente Form possui alguns elementos/componentes filhos, a exemplo do que acontece no HTML comum. Entretanto, aqui há alguns elementos diferentes, específicos da implementação em questão, como o <Form.Group>, o <Form.Control>, entre outros. Além disso, perceba que nos componentes são definidas algumas classes, como "mb-3", por exemplo. Essa classe é usada pelo React Bootstrap para definir uma "margin-bottom" no valor de 3. Além desta, há várias outras classes pré-definidas disponíveis. Ainda sobre o Formulário, há outras opções de configuração do layout, como formulários on-line, por exemplo. Como sempre, a documentação oficial deve ser consultada para a obtenção de mais detalhes, exemplos de código e informações.

Após uma breve introdução sobre o react-bootstrap, sabemos como ele pode nos ajudar na construção de nosso Projeto Integrado, ou seja, fornecendo componentes visuais e funcionais que permitirão

aprimorar a experiência dos usuários ao utilizarem a aplicação, além de possibilitar a construção de páginas responsivas. A partir daí, colocaremos a mão na massa e modificaremos o CRUD de tarefas para utilizar o React Bootstrap, no lugar do Material UI. O código resultante dessa modificação pode ser conferido no link abaixo. Contudo, antes de olhar o código, tente fazer sozinho essa modificação. Mais uma observação: precisaremos instalar uma outra biblioteca, para incluirmos ícones em nossas telas. Para isso, usaremos a react-icons (embora você possa utilizar outra de sua preferência). Para instalá-la, execute o comando abaixo:

```
npm install react-icons
```

Link para o repositório contendo a aplicação, modificada para utilizar o react-bootstrap:

https://github.com/FaculdadeDescomplica/pratica_integradora_tecnologias_disruptivas/tree/main/modulo11/template-bootstrap **(Acesso em 29/09/2022)**

Atividade Extra

Ao longo desse módulo e também em sua atividade prática, nos deparamos com um novo framework, repleto de recursos como diversos componentes, padrão de grid, forma de estilização, etc. Para melhor aproveitarmos todos esses recursos, é necessário

olharmos a documentação oficial, onde temos acesso aos detalhes e também a exemplos de código. Nesse sentido, sugiro que navegue pela documentação oficial do React Bootstrap e veja, sobretudo, os exemplos de código disponíveis.

Link: https://react-bootstrap.github.io/ **(Acesso em 29/09/2022)**

Referência Bibliográfica

MORGAN, Joe. **How to code in React.js.** Digital Ocean, 2021.

Atividade Prática Módulo 11

Título da Prática: Adaptando o projeto template.

Objetivos: Praticar a codificação React através da refatoração de código.

Materiais, Métodos e Ferramentas: IDE VS Code.

Atividade Prática

Ao longo do Módulo 11, vimos uma nova biblioteca. Nesse caso, um framework, a React Bootstrap. Tal framework foi utilizado na estilização de nossa aplicação template, inicialmente desenvolvida com a lib Material UI. Durante esse processo, as telas de Listagem e Criação de Tarefas foram refatoradas: os componentes do Material UI foram substituídos por componentes do React Bootstrap. Agora,

você deverá finalizar essa refatoração, modificando a página de Editar Tarefa, que não foi disponibilizada no repositório, junto com as demais páginas. Além disso, você deverá descomentar o código, na Listagem, para que a tela de edição possa funcionar.

O passo a passo abaixo vai orientá-lo para a realização dessa atividade:

1. Clone o projeto localmente, criando assim um novo projeto React;

2. Analise o componente CriarTarefa.js, já refatorado para utilização do React Bootstrap no lugar do Material-UI;

3. Crie o componente EditarTarefa.js, tendo como base o mesmo componente utilizado no Módulo 10, modificando-o conforme passo anterior;

4. Ao final, teste o funcionamento da aplicação e versione seu código.

Gabarito Atividade Prática

Para a realização dessa atividade são descritos alguns passos para orientar o aluno. Ao final de todo o processo, ele deverá ter versionado seu código, modificando o componente ListarTarefa.js e criando o componente EditarTarefa.js. O código desses dois componentes estão disponíveis no link abaixo. Considerando que o aluno poderá realizar suas próprias customizações, deverá ser

analisado, sobretudo, se ele foi capaz de remover do projeto a biblioteca Material-UI, incluir a React Bootstrap e deixar o projeto funcional.

Link:
https://github.com/FaculdadeDescomplica/pratica_integradora_tecnologias_disruptivas/tree/main/modulo11/atividade-pratica-gabarito **(Acesso em 28/09/2022)**

Ir para exercício